# **PART 0. INTRODUCTION (VINH)**

1. Introduce about your recent project, role, and responsibility?

* Introduce about “difficult”, “challenging” project first, **don’t care timeline**.
* For each project, need to introduce:
  + Small description about project’s business, don’t care project name (because as usual project name is just an acronyms or doesn’t make sense).
  + Technologies was used in the project.
  + Your responsibilities (not manager, need to be a key person)

1. Can you explain the JavaEE experiences in your worked project?

* Need to mention some techniques: RMI, Web Service (SOAP, RESTful), JSP/Servlet, …
* How did you apply the techniques into your projects?
* How many year have you been worked with J2EE?

1. Could you please introduce yourself and tell us more about work experience?

You need to outline your ideas into some items:

* Part 1: Name (short name)
* Part 2: Number of year experience
* Part 3: Experiences (framework, technologies, techniques, difficult solved-problem).
* Part 4: The most exciting projects (choose the best one). Tell about techniques and solution for this project.
* Part 5: Motivation.

# **PART 1. BASIC (DONNIE)**

1. What equals() and hashCode() method respond for? How and when override them?

* Open equals and hashcode function in Object.java source code.
* Understand how hashcode and equals are used in HashMap by opening HashMap source code.
* Understand contract between equals and hashcode function.

1. When you override hashCode, which method do you override?

* Equals, why? 🡺 refer to contract between hashcode and equals function (open source code of HashMap to view)

1. If two objects are not equals by equals() method 🡪 Is it require that hashcode must be the same?

* Possible, hashcode can be the same eventhough equals = false. (base on contract between hashcode and equals function)

1. What are differences between Deep copy and Shallow copy?

* Deep copy: clone object and related child object
* Shalow copy: just copy referfence address of an instance.

1. Is String mutable? Why?

* String is immutable. Because String need to be a thread-safe object in order to process for many threads and considered as a constant for many formulas.

1. What finally() block use for?

* To release/clean up resource after using or after throwing exception.

1. What are pass by reference and pass by value?

* Pass by reference means the called functions parameter and the callers passed argument are the same.
* Pass by value means the called functions parameter is a copy of the callers passed argument.
* In Java, It is pass by value. Java passes objects as references and those references are passed by value.

1. What is the difference between a constructor and a method?

* Constructor, member function, used to create objects of that class.
* Method, ordinary member function of a class, has own name, a return type

1. What if the main method is declared as private?

* Will compile, but it will not run.

1. What if the static modifier is removed from the signature of the main method?

* Will compile, but it will not run.

1. What id I write “static public void” instead of “public static void”?

* Will compile, and will run.

1. What if I don’t provide the String array as the argument to the method?

* Will compile, but it will not run.

1. What is the first argument of the String array in main method?

* Will compile, but it will not run.

1. If I do not provide any arguments on the command line, then the String array of Main method will be empty or null?

* Empty

1. How to print “Hello World” with only one line of code in Java?
2. What environment variables do I need to set on my machine in order to be able to run Java program?
3. Can an application have multiple classess having main method?

* Yes

1. Can I have multiple main methods in the same class?

* No

1. Do I need to import java.lang package any time? Why?

* No

1. Can I import same package/class twice?Will the JVM load the package/class twice at runtime?

* Yes and No

1. Does importing a package imports the subpackages as well? E.g. Does importing com.MyTest.\* also import com.MyTest.UnitTests.\*?

* No, only classes on same package. Subpackage NO.

1. What is the difference between declaring a variable and defining a variable?

* String str; <- Declaring
* String str = "Hello"; Declaring and Defining.

1. What is the default value of an object reference declared as an instance variable?

* null

1. Can a top level class be private or protected?

* public, abstract & final are allowed modifiers for a class.

1. What type of parameter passing does Java support?

* Pass By Value

1. Primitive data types are passed by reference or pass by value?

* Pass By Value

1. Object are passed by value or by reference?

* Pass By Value

1. Give a simplest way to find out the time a method takes for execution without using any profiling tool?

* Get time difference.

1. What are wrapper classes? Why do we need wrapper classes?

* A class that wraps primitive

1. What is the difference between the instanceof and getClass, these two are same or not?

* instanceOf, operator
* getClass(), method

1. What are static block?

* Executed first when an object is instantiated

# **PART 2. EXCEPTION HANDLING (DONNIE)**

1. Is it necessary that each try block must be followed by a catch block?

* No. Either Catch, Finally or Both.

1. What are Checked and UnChecked Exception?

* Checked, compile time
* Unchecked, runtime

1. What are runtime exceptions?

* Exceptions occurred at runtime

1. What is the difference between error and an exception?

* Error - Unrecoverable
* Exception - Recoverable

1. How to create custom exception?

* extends Exception

1. What are the different ways to handle exceptions?

* Try, Catch, Throws

1. What is the basic difference between the 2 approaches to exception handling.  
   1> try catch block and (You handle the problem)  
   2> specifying the candidate exceptions in the throws clause? (You let them handle the problem)  
   When should you use which approach? Try - owning responsibility. Throws when making libraries or let them own the responsibility.
2. If I write return at the end of the try block, will the finally block still execute?

* Yes

1. If I write System.exit (0); at the end of the try block, will the finally block still execute?

* No

# **PART 3. Java OOP (VINH)**

1. Compare Object vs Class vs Instance?

* An object is a software bundle of related state and behavior.
* A class is a blueprint/template or prototype from which objects are created.
* An instance is a single and unique copy of a class that representing an Object.

1. Describe the principles of OOP? Explain for each principle?

* Encapsulation:
  + encapsulate states, behaviors in a class, abstract class or interface;
  + restrict access by access modifiers.
* Abstraction: abstract states/behaviors of an object by using abstract class, interface, extends, super, this.
* Inheritance: extends states, behaviors of a super class.
* Polymorphism: abstract class, interface, overriding, overloading.

1. How many way to implement inheritance in Java?

* Extends: ChildClass extends ParentClass.
* Delegation: ChildClass contain one attribute is an instance of ParentClass.

1. Why multiple inheritance is not supported in Java?

* Ambiguity around Diamond problem

1. Expplain the different forms of Polymorphism.

* Runtime Polymorhism( or Dynamic polymorphism): Overriding
* Compile time Polymorhism( or Static polymorphism): Overloading

1. What is an abstract class?
2. What is an Interface?
3. What is static in java?
4. What is final in Java? How is it used?
5. What is method overloading, method overriding? What are the differences?
6. How to prevent a method from being overridden?

* final

1. Do interface have member variables?

* No. Why?

1. What midifiers are allowed for methods in an Interface?

* public, abstract

1. How to override the main method?

* No

1. How to invoke a superclass version of an overridden method?

* Use super.method()

1. Compare Composition, Aggregation, Association vs Inheritance?

* Inheritance: Manager is an employee of XYZ limited corporation.
* Association: Manager uses a swipe card to enter XYZ premises.
* Aggregation: Manager has workers who work under him.
* Composition: Manager has the responsibility of ensuring that the project is successful.
* Composition: Manager's salary will be judged based on project success.

1. What is the difference betweeen an Interface and an Abstract class?

* Different in syntax:…
* Different in usage:
  + Interface: for optional behavior, API definition.
  + Abstract class: for some common behavior, all sub-class need to have.

1. When should I use Abstract Class, when shoud I use Interface?

* Give 3 examples base on answer in previous questions.
  + Example about Abstract Class
  + Example about Interface
  + Example about combination of Abstract class and Interface.

1. Give an example of Interface in real life?
2. Give an example of Abstract Class in real life?
3. Give an example of Abstract Class combine with Interface?
4. State the significance of public, private, protected, default modifiers?
5. What is Overriding?
6. What are different types of inner classes?
7. What is a marker Interface?

* Interface doesn’t have anything

1. Can we declare an abstract method in a normal class?

* Impossible. Why? (what happen if people invoke the abstract method of the instance which instantiate from this normal class)

1. How does the Java default constructor be provided?

* If you don’t define your own constructor.

1. Can constructor be inherited?

* Yes, use super(…)

1. What are the differences between Constructors and Methods?

* Syntax: constructor doesn’t have return type, name = class name.
* Usage: constructor is used to instantiate an object.

1. How are this() and super() used with constructors?
2. What are the diffrences between Class Methods and Instance Methods?

* Class method: static method
* Instance method: normal method

# **PART 4. SERIALIZATION (DONNIE)**

1. What is serialization and Why serialize?

* Transformation of objects to bytes. Save to file, database or send it over the network.

1. How do I serialize an object to a file?

* Implement Serializable. Use ObjectOutputStream.

1. Which methods of Serializable interface should I implement?

* None

1. How can I customize the serialization process? i.e how can one have a control over the serialization process?

* Externalizable

1. What is the common usage of serialization?

* Data transfer

1. What is Externalizable interface?

* Custom serialization process

1. When you serialize an object, what happens to the object references included in the object?
2. What one should take care of while serializing the object?

* Make sure that objects included are also serializable

1. What happens to the static fields of a class during serialization?

* Not included.

1. What are serialization and externalization?
2. What version Id when serializing use for?

* serialVersionUID

1. What is reflection?

* To describe code which is able to inspect other code in the same system

1. What is transient variable?

* Not include during serialization process

# **PART 5. COLLECTION (DONNIE)**

1. What is Collection API?

* Set of classes and interface that allows you to store objects in a collection.

1. How do you traverse through a collection?

* Loops, Iterator

1. What is the List interface?

* Ordered collection according to insertion

1. What are the main implementations of the List interface?

* ArrayList, LinkedList, Vector

1. What is the Set interface?

* No duplicates

1. How to remove the duplication of the Set Collection?

* Nothing to remove.

1. How to remove the duplication of the List Collection?

* Copy to Set then copy back to List.

1. What are the main implementations of the Set interface?

* HashSet, TreeSet, LinkedHashSet

1. What are differences between HashSet and TreeSet?

* Hashset = not sync, unordered, allow NULL.
* Treeset = not sync, natural order, faster

1. What is a Map?

* Key Value Pair

1. What are the main implementations of the Map interface?

* HashMap, TreeMap, Hashtable

1. How do you sorting a list of user-defined objects?

* Comparable, Comparator

1. What are the differences between the Comparable and Comparator interfaces?

* Comparable, compareTo : Comparator, compare

1. What is difference between Arrays and ArrayList ?

* Arrays:FixedSize, primitives, Objects, multidimensional. ArrayList:Growable, Objects only, single dimension

1. What are the advantages of ArrayList over arrays?

* Lot of helper methods and dynamic.

1. What are differences between ArrayList and Vector?

* Unsynchronized and Synchronized.

1. What are differences between HashMap and HashTable?

* Hashmap allows NULL while HashTable does not, and HashMap is not synchronize while Hashtable is synchronized.

1. What are differences between LinkedList and ArrayList?

* ArrayList uses index to access elements, faster than LinkedList, it is more lighter because memory address only contains the value. LinkedList is using pointers to next and previous elements

1. How do you decide when to use HashMap and when to use TreeMap?

* Use HashMap if you don't care of the order of the elements, and use TreeMap if you want your elements to be sorted

1. What are differences between HashMap and TreeMap?

* Hashmap is implemented as a hash table, and unordered while TreeMap is ordered

1. What is Set interface? TreeSet?
2. What is Iterator? How to use it? When you use For loop, when you use Iterator?
3. Why are Iterators returned by ArrayList called Fail Fast?
4. What are differences between List and Set?
5. Do List & Set have common parent?

* Yes. Collection.

1. Is Iterator a Class or Interface? What is its use?

* Interface, it is used to traverse the collection.

1. What is Collection.synchronizeList()?

* make List synchronized. It is use to make non-thread safe collection to be thread-safe

# **PART 6. GARBAGE COLLECTION (DONNIE)**

1. What is the purpose of garbage collection in Java, and when is it used?

* To removed unreferenced Objects.

1. How to do GC tunning?
2. How many type of memory in JVM?

* Heap Memory - storage for Java Objects
* Non-Heap Memory - stores loaded class and other metadata

1. What is young-generation, old-generation memory?

* Young-generation – composed of one eden space and two survivor spaces. The place where all new objects are created.
* Old-generation – contains the objects that are long lived and survived many rounds of Minor GC.

1. What is isolated-island in term of GC?

* Describes one or more objects having no reference to them from active parts of an application.  
  When Object A references Object B and Object B references A but no other Objects references both.

1. When object will be remove by GC?

* If the object cannot be reached by any live thread.

1. Can you force GC to remove an object?

* No

# **PART 7. MULTITHREADING (VINH)**

1. Describe synchronization in respect to multithreading
2. Explain different way of using thread?

* Extends Thread
* Implements Runnable

1. How to implement Java thread?

* Override run() method

1. What is thread safe?

* Take time to read this: <https://en.wikipedia.org/wiki/Thread_safety>

1. How to implement thread safe?

* Immutable class
* Semaphore
* Synchronization

1. What is deadlock?

* Circular waiting for resource.

1. What is thread monitor?

* Use some tool like: Jconsole, JvisualVM to monitor Thread.

1. What are Critical sections, Mutex (Mutual Exclusion) and Semaphore?

* Critical sections: road intersections. Vehicles move in different directions(different threads) so these intersections need traffic lights(synchronize) to avoid accidents.
* Mutual exclusion: If in your house there is only one restroom, once someone uses it, the others must wait for you to finish your business there before using it.
* Semaphore: You are in a restaurant with limited number of tables. The reception lady will act as a semaphore. You will only be accommodated when there is an empty table. if there's none, you have to wait after others will finish eating.

# **PART 8. PERFORMANCE TUNING (VINH)**

1. How to monitor JVM performance?

* Use Jconsol, jVisualVM, logging CPU+Memory+TCP open connection, GC (overhead?)

1. How to do JVM tuning?

* Increase headsize
* Increase perm size
* Configure how long for GC to collect unused variables.

1. Have you ever faced to performance issues and how to solve them?

* State your own problem and your solution. If you don’t have, skip this question. But, this is IMPORTANT question. Your assessment is very high if you can answer this question.

# **PART 9. SPRING (GLENN)**

1. How many modules in Spring Framework?
2. How many modules do you work with?
3. What is IOC ? How to inject in spring bean?
   1. (discuss inversion of control)
4. Give me 5 reasons to use spring.
   1. (keywords: easier, dependencies, modularity, Flexible)
5. What is IoC vs DI?
   1. (discuss difference of IoC and dependency injection)
   2. (also check about the other forms of IoC)
6. What is AOP?
   1. (discuss aspect-oriented programming)
   2. (discuss how to implement)
7. What are the different modules in Spring framework?
8. What are important ApplicationContext implementations in Spring framework?
9. What is BeanFactory interface?
10. Do I need to instantiate the container more than once? Why?
11. Can you integrate multiple bean configuration xml?
12. How many are the different types of AutoProxying, What are they?
13. What are the 5 types of AOP Advice?
    1. (also provide examples of practical uses)
14. What is the difference between singleton and prototype bean?
    1. (also research about bean scopes)
15. What kind of exceptions does spring DAO classes throw?
16. How many methods to provide configuration metadata to the Spring Container?
    1. (also know how to implement)
17. What is an Aspect?
18. What is a Jointpoint v/s Advice v/s Pointcut?

# **PART 10. REST – WEB SERVICE (GLENN)**

1. How many year of restful web service experience do you have?
2. How do you use a web service?
   1. (include discussion about input/output, content format)
3. Have you worked with web services before? (should explain SOAP or REST)
4. How would you implement a rest web service? (should explain SOAP or REST)
   1. use the following keywords in your discussion
      1. Spring Web MVC
      2. DispatcherServlet
      3. Presentation Layer
      4. Service Layer
      5. Persistent Layer
      6. Annotations
5. How can we parse Json and XML with REST?
   1. (discuss Jackson and JAXB)
6. Can you explain how to implement Restful by using Spring framework?
   1. (cite the libraries needed, discuss the components you need to create)
7. How can implement the restful web service to return the json , xml , you can use any framework?
   1. (discuss Content-type Negotiation)

<http://theblasfrompas.blogspot.com/2013/10/spring-mvc-rest-content-negotiation.html>

1. If I want a json response or xml response, what’s difference between requests?
2. Could you give me some steps to create a restful Web Service?
   1. use the following keywords in your discussion
      1. Spring Web MVC
      2. DispatcherServlet
      3. Presentation Layer
      4. Service Layer
      5. Persistent Layer
      6. Annotations
3. What are differences between GET and POST method? How about other method?
   1. (mention the supported length, data type, resource, cache)
4. What is REST and RESTful web services ?
   1. (discuss in terms of communication, state, cache)
5. What is differences between RESTful web services and SOAP web services ?
   1. (discuss in terms of protocol, uses, format, performance, cache)
6. What is Restlet framework ?
7. What is Resource in REST framework ?
8. Can you use Restlet without any web-container ?
9. What is difference between Restlets and Jersey ?
10. What is RESTEasy ?
11. What are the tools used for creating RESTFull web services ?
12. How to display custom error pages using RestFull web services ?
13. Which HTTP methods are supported by RestFull web services ?
14. What is difference between top-down and bottom-up approach of developing web services ?
15. What happens if RestFull resources are accessed by multiple clients ? do you need to make it thread-safe?
16. What differences between synchronous vs asynchronous web services?
17. What is WSDL?
    1. (keywords: web service, operations, parameters)
18. What is JAX-WS and JAX-RS?
19. What are the underlying protocol over which you can send SOAP request?
20. What are the different formats supported by REST API?
21. What is use of JAX-RPC?
22. How many types of JAXM messaging models?
23. Explain WSDL types?
24. REST vs SOAP?
25. What is UDDI?
26. @Service, @Controller, @Repository, @Component?
27. What kind of design pattern used in Spring?
28. What is used for parsing XML and JSON in Spring?
29. What is JAXB?
30. What are differences between DOM parser and SAX parser?
31. What Maven used for?
    1. (keywords: build automation)
32. What is POM file?
    1. (keywords: configurations, dependencies)
33. Do you use maven? Multi modules or single module?

# **PART 11. DATABASE – JDBC – HIBERNATE (GLENN)**

1. What is outer join, left join, inner join?
2. What is index? When should you use index?
3. What is transaction? Why do you use transaction?
4. Explain the new Features of JDBC 2.0 Core API?

# **PART 12. DESIGN PATTERN (VINH)**

1. What kind of design pattern you know?

* Singleton
* Factory
* Visitor
* Observer
* Façade
* Template
* Service Locator
* Front Controller
* IoC
* MVC
* Iterator
* Comparable vs Comparator

1. What façade pattern, factory pattern, singleton, observer, IoC, Dependency Injection, AOP pattern? When you use them?
2. What is service locator pattern?
3. What is an Iterator?

# **PART 14. DESIGN (VINH)**

1. Which tool do you use to draw diagram? Which kind of diagram do you use?

* Sequence diagram: Visio, Rational Rose.
* Activity diagram: Visio
* System Integration diagram: Visio, Power Point, Excel, Word

# **PART 15. XML (GLENN)**

1. What is XML Schema?
   1. keyword: structure
2. What is JAXP, JAXB?
   1. also discuss differences
3. If we have convert objects from Java to XML, what the framework you will use?

# **PART 16. TESTING (GLENN)**

1. How to write a test case?
2. Have you used unit testing before?
3. When to use @Before and when to use @After in unit tests?
4. What test framework do you use in your project?
5. What junit life circle?
   1. (also mention the annotations)
6. How to implement the method test?
7. Do you know some automation test tool ?
8. How to implement the test the restful web service ?
9. Could you give me the life-cycle of unit test?
10. Do you have experience about Mockito & Power Mockito?

<http://www.vogella.com/tutorials/Mockito/article.html>

<http://www.rapidprogramming.com/questions-answers/difference-between-mockito-and-powermock-mockito-vs-powermock-1504>

1. How to test web service with the data will be updated everyday?
   1. (discuss the test case for this)
2. Can you give me an example when we use before and after?
3. If we have to write a test case to test the method convert Java object to XML (ex: Employee object) using JAXB, how do we implement/arrange the test case?
4. If we have to write a test case to test a restful web service, how do you write in java?

# **PART 17. CONCLUSION (VINH)**

1. **Do you have any question for us? (IMPORTANT, LAST IMPRESSION)**

* What technologies will be used in your project?
* Do I need to research about some technologies?
* Which kind of caching methodology that you prefer?
* What I can achieve while working with your project? (need to confidence)